

# **PRAKTIKUM PEMROGRAMAN FRAMEWORK**

## **MODUL 11 REST API LARAVEL**



**Sistem Informasi  
Telkom University  
Surabaya**

Disusun Oleh:

Tim Asisten Praktikum Pemrograman Framework

**PROGRAM STUDI S1 SISTEM INFORMASI**

**FAKULTAS REKAYASA INDUSTRI**

**TELKOM UNIVERSITY SURABAYA**

**2024**

## DAFTAR ISI

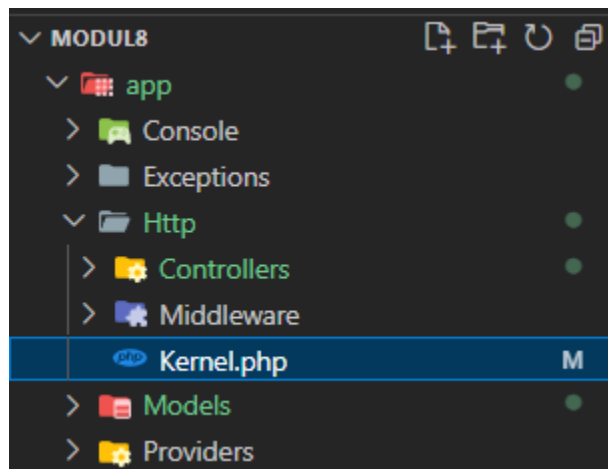
1. Mengintegrasikan laravel dengan API.....	2
2. Membuat Controller untuk mengintegrasikan register dengan API.....	3
3. Mengatur Route untuk mengintegrasikan API.....	4
4. Mengintegrasikan POSTMAN.....	5

# MODUL

## REST API LARAVEL

### 1. Mengintegrasikan laravel dengan API

- Untuk mengintegrasikan Laravel dengan API, perlu menambah beberapa konfigurasi, salah satunya yaitu memodifikasi file kernel pada Laravel
- Buka file kernel pada project Laravel pada direktori **app/Http/kernel.php**



- Modifikasi file **kernel.php** pada bagian **api** dengan menambahkan kode berikut

```
\Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,  
        'throttle:api',  
  
\Illuminate\Routing\Middleware\SubstituteBindings::class,
```

- Berikut tampilan file **kernel.php** ketika sudah di modifikasi

```
protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],

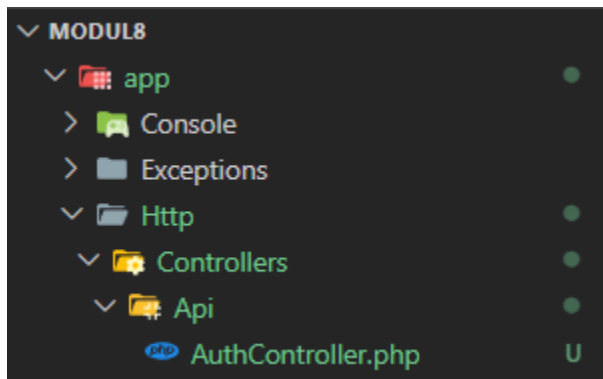
    'api' => [
        \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
        'throttle:api',
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],
];
```

## 2. Membuat Controller untuk mengintegrasikan register dengan API

- Membuat Controller Auth di terminal dengan command berikut  
**php artisan make:controller Api\AuthController**

```
PS D:\File Kuliah\ASPRAK\PF 05-01\modul8> php artisan make:controller Api\AuthController
INFO Controller [D:\File Kuliah\ASPRAK\PF 05-01\modul8\app\Http\Controllers\Api\AuthController.php] created successfully.
```

- Setelah membuat controller tersebut maka buka file controller pada direktori App/Http/Controllers/API/AuthController



- Terapkan kode berikut pada file tersebut untuk melakukan register dengan metode API

```
<?php

namespace App\Http\Controllers\API;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Auth;
```

```

use Illuminate\Support\Facades\Validator;

class AuthController extends Controller
{
    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'name' => 'required',
            'email' => 'required|email',
            'password' => 'required',
            'confirm_password' => 'required|same:password'
        ]);

        if ($validator->fails()) {
            return response()->json([
                'success' => false,
                'message' => 'Ada kesalahan',
                'data' => $validator->errors()
            ]);
        }

        $input = $request->all();
        $input['password'] = bcrypt($input['password']);
        $user = User::create($input);

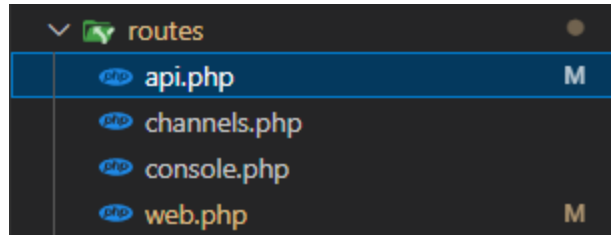
        $success['token'] =
        $user->createToken('auth_token')->plainTextToken;
        $success['name'] = $user->name;

        return response()->json([
            'success' => true,
            'message' => 'Sukses register',
            'data' => $success
        ]);
    }
}

```

### 3. Mengatur Route untuk mengintegrasikan API

- Buka file **api.php** pada direktori **routes/api.php**



- Tambahkan kode berikut pada route api untuk mengintegrasikan route untuk register melalui API

```
Route::post('register', [AuthController::class, 'register']);
Route::post('login', [AuthController::class, 'login']);

Route::middleware('auth:sanctum')->get('/user', function
(Request $request) {
    return $request->user();
});
```

- Berikut tampilan file **api.php** ketika sudah di modifikasi

```
routes > api.php > ...
1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5  use App\Http\Controllers\API\AuthController;
6
7  /*
8   |-----
9   | API Routes
10  |-----
11  |
12  | Here is where you can register API routes for your application. These
13  | routes are loaded by the RouteServiceProvider within a group which
14  | is assigned the "api" middleware group. Enjoy building your API!
15  |
16  */
17
18 Route::post('register', [AuthController::class, 'register']);
19 Route::post('login', [AuthController::class, 'login']);
20
21 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
22     return $request->user();
23 });
24
```

#### 4. Mengintegrasikan POSTMAN

- Buka website **postman** untuk melakukan integrasi **REST API**, berikut link website nya <https://www.postman.com/downloads/>
- Klik download pada web postman

upportPublic API Network

Contact SalesLaunch Postman

## Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

### The Postman app

Download the app to get started with the Postman API Platform.

Windows 64-bit

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

[Release Notes](#)

Not your OS? Download for Mac ([Intel Chip](#), [Apple Chip](#)) or Linux ([x64](#), [arm64](#))

### Postman on the web

Access the Postman API Platform through your web browser. Create a free account, and you're in.

Launch Postman

HomeWorkspacesAPI Network

Search Postman

Get started

Notion's Public WorkspaceNewImport

Overview

retrieve a database

Collection

Notion API

Users

Databases

Retrieve a database

200 Success- Retri...

Query a database

Create a database

Update a database

Pages

Users

Blocks

Comments

GET

https://api.notion.com/v1/databases/id

Send

ParamsAuthHeaders(TO)BodyScriptsSettings

Query params

KEYVALUEDESCRIPTIONBulk edit

KeyValueDescription

Path variables

KEYVALUEDESCRIPTIONBulk edit

id(DATABASE\_ID)Required. Enter database id.

Body

200 OK1033ms120KBSave as Example

PrettyRaw

1{

2}

3}

4}

5}

6}

Documentation

https://api.notion.com/v

Retrieves a database obje specified in the request p

AuthorizationBearer

This request is using an a header from collection Not

Request Header

Notion-Version22-0

Path Variables

id

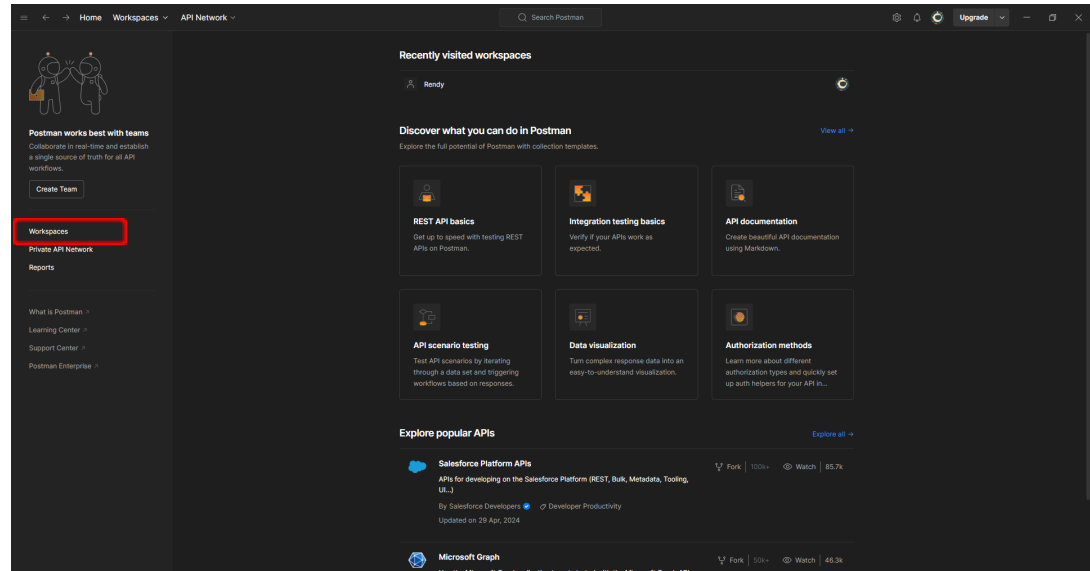
We use cookies on your device to enhance your navigation experience, analyze usage to improve our site, and customize our marketing efforts. You can learn more about cookies in our [Cookie Notice](#).

Manage Cookies

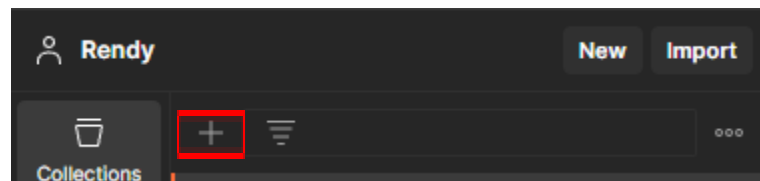
Reject All Cookies

Accept All Cookies

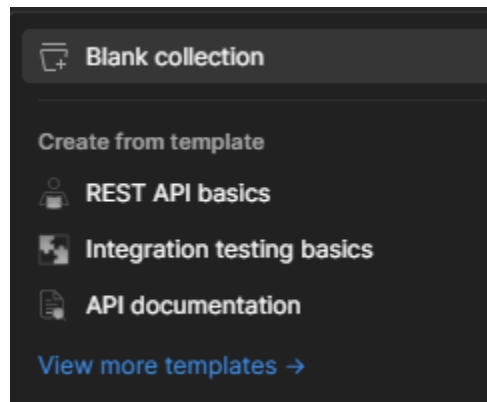
- Setelah itu install postman seperti biasa
- Buat workspace baru dengan nama bebas



- Kemudian klik icon tambah

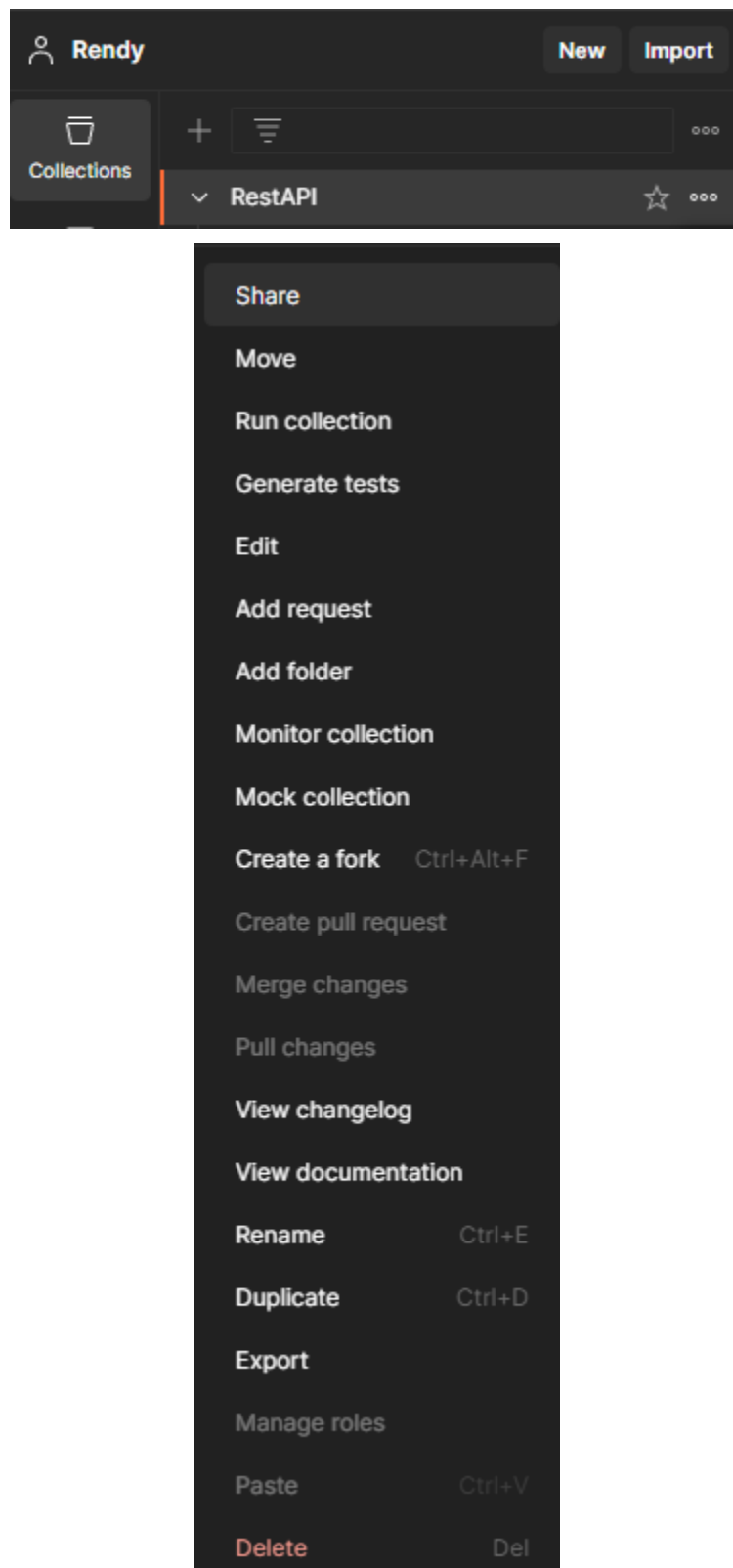


- Pilih blank collection dan beri nama bebas untuk praktikum “RestAPI”

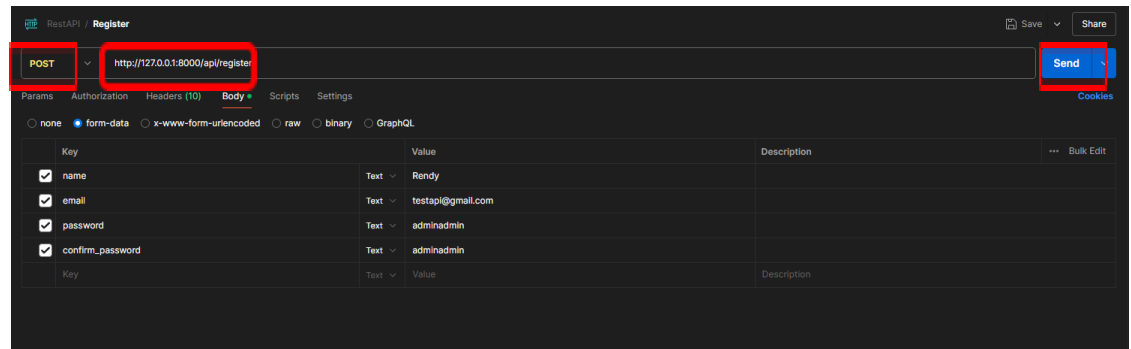




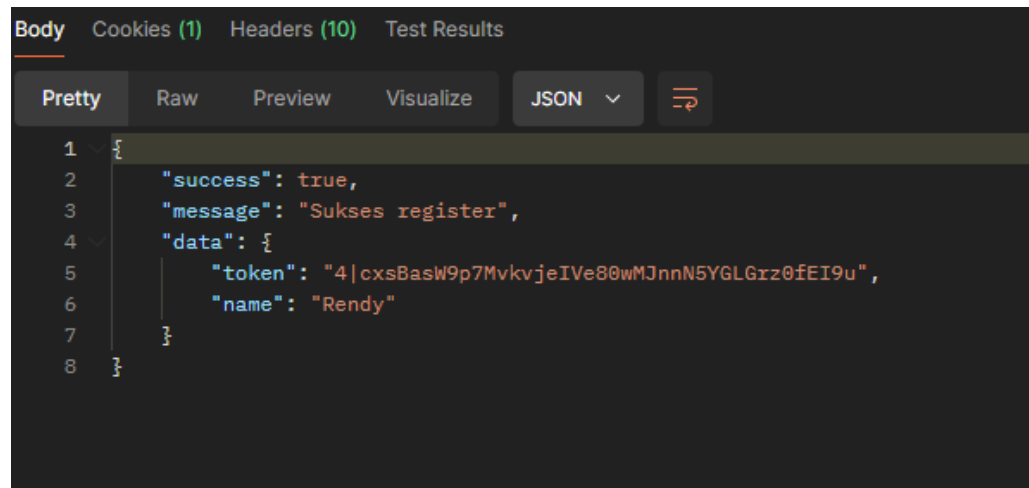
- Klik titik 3 dan pilih add request



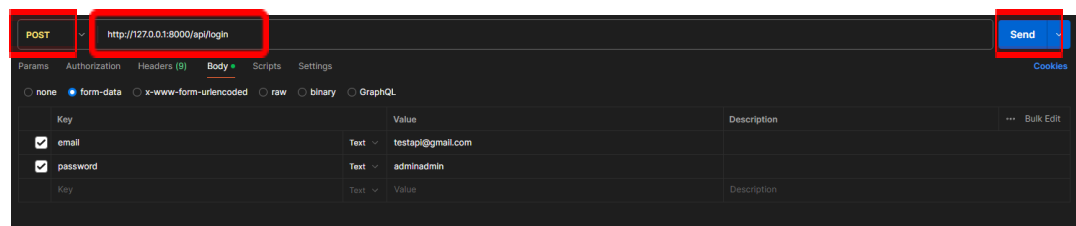
- Kemudian sesuaikan pada api untuk register seperti dibawah, kemudian jalankan dengan klik send



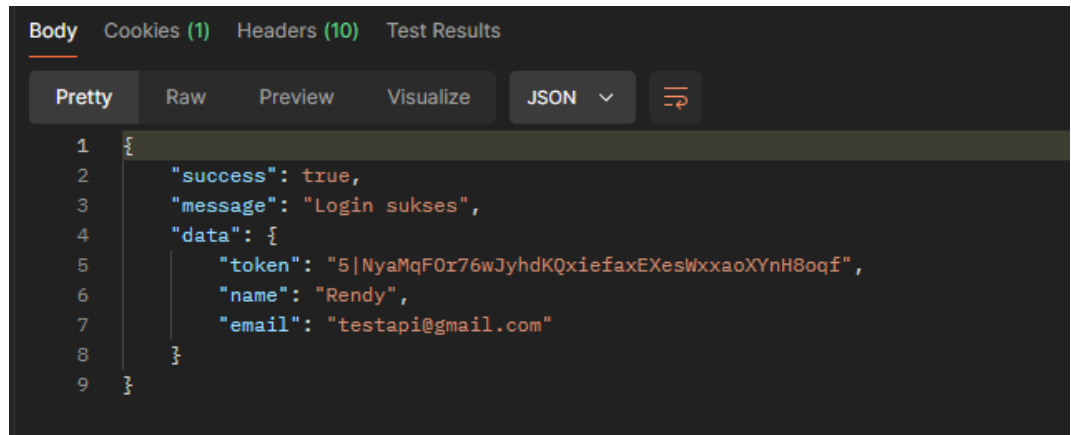
Jika berhasil maka tampilan outputnya seperti berikut:



- Kemudian buat add request untuk login, sesuaikan pada api untuk login seperti dibawah, kemudian jalankan dengan klik send



Ketika berhasil login maka tampilan outputnya seperti berikut



The screenshot shows a web browser's developer console with the 'Body' tab selected. The response is displayed in a 'Pretty' JSON format. The JSON object contains a 'success' field set to true, a 'message' field with the text 'Login sukses', and a 'data' field which is an object containing 'token', 'name', and 'email' fields. The token is a long alphanumeric string, the name is 'Rendy', and the email is 'testapi@gmail.com'.

```
1 {  
2   "success": true,  
3   "message": "Login sukses",  
4   "data": {  
5     "token": "5|NyaMqF0r76wJyhdKQxiefaxEXesWxxaoXYnH8oqf",  
6     "name": "Rendy",  
7     "email": "testapi@gmail.com"  
8   }  
9 }
```